Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1 - 31. (Canceled)

32. (Previously presented): A method for accessing a datum stored in a memory comprising:

accessing the memory to read out a first portion of the datum, the first portion comprising one or more first bytes;

performing a first rotation operation on the first bytes to produce rotated first bytes, wherein a rotate distance of the first rotation operation is based on whether the datum stored in the memory is stored in Big Endian order or in Little Endian order;

replacing portions of the rotated first bytes with binary zeroes to produce aligned first bytes;

storing the aligned first bytes to a first data store;

accessing the memory to read out a second portion of the datum, the second portion comprising one or more second bytes;

performing a second rotation operation on the second bytes to produce rotated second bytes, wherein a rotate distance of the second rotation operation is based on whether the datum stored in the memory is stored in Big Endian order or in Little Endian order;

replacing portions of the rotated second bytes with binary zeroes to produce aligned second bytes;

storing the aligned second bytes to a second data store; and

performing an OR operation of the content of the first data store and the second data store to produce the datum.

33. (Previously presented): The method of claim 32 wherein the datum as stored in the memory can be an aligned datum or an unaligned datum.

34. (Previously presented): The method of claim 32 wherein the step of replacing portions of the rotated first bytes with binary zeroes includes performing a masking operation.

35. (Previously presented): The method of claim 34 wherein the step of replacing portions of the rotated second bytes with binary zeroes includes performing a sign extension operation.

36. (Previously presented): The method of claim 32 wherein the datum is one of a 16-bit datum, a 32-bit datum, and a 64-bit datum.

37. (Previously presented): The method of claim 32 further comprising storing the datum to a third data store subsequent to performing the OR operation.

38 - 41. (Canceled)

42. (Previously presented): A CPU having a circuit configured to access a datum stored in a memory comprising:

a data input for receiving a portion of the datum from the memory;

a rotator circuit coupled to the data input and having a rotator output, the rotator circuit configured to perform a rotation operation of data on the input and to produce rotated data at the output;

a distance control circuit having a control signal coupled to the rotator circuit, wherein the control signal is dependent on whether the datum that is stored in the memory is stored in Big Endian order or in Little Endian order, wherein an amount of rotation performed by the rotator circuit is dependent on the control signal; and

a zero-filling circuit coupled to the rotator output and configured to replace one or more bit positions of a datum on the rotator output with binary zeroes.

43. (Previously presented): The circuit of claim 42 wherein the datum stored in the memory can be an aligned datum or an unaligned datum.

44. (Previously presented): The circuit of claim 42 wherein the datum stored in the memory comprises a first portion of one or more bytes of data and a second portion of one or more bytes of data, wherein the first portion is processed by the circuit to produce a first result and the second portion is processed by the circuit to produce a second result, wherein an OR operation performed on the first result and the second result produces the datum stored in the memory.

45. (Canceled)

46. (Previously presented): A data processor system comprising:

a memory, wherein a first datum is stored in the memory, wherein the first datum can be an aligned datum or an unaligned datum;

a data input for receiving a portion of the first datum from the memory;

a rotator circuit coupled to the data input and having a rotator output, the rotator circuit configured to perform a rotation operation of data on the input and to produce rotated data at the output;

a distance control circuit having a control signal coupled to the rotator circuit, wherein the control signal is dependent on whether the first datum is stored in the memory in Big Endian order or in Little Endian order, wherein an amount of rotation performed by the rotator circuit is dependent on the control signal; and

a zero-filling circuit coupled to the rotator output and configured to replace one or more bit positions of a datum on the rotator output with binary zeroes.

47. ((Previously presented): The data processor system of claim 46 wherein the datum stored in the memory comprises a first portion of one or more bytes of data and a second portion of one or more bytes of data, wherein the first portion is processed by the circuit to produce a first result and the second portion is processed by the circuit to produce a second result, wherein an OR operation performed on the first result and the second result produces the datum stored in the memory.